Debugging/Profiling w/ Julia

Options:

- Debugging
  - VSCode's debugger
    - https://www.julia-vscode.org/docs/stable/userguide/debugging/
  - Debugger.jl
    - https://github.com/JuliaDebug/Debugger.jl
  - Infiltrator.jl
    - https://github.com/JuliaDebug/Infiltrator.jl?tab=readme-ov-file
- Profiling
  - @code_warntype
  - @profview
  - @profview_allocs
  - CUDA.@profile
    - https://cuda.juliagpu.org/v2.2/development/profiling/
  - Nvidia Nsight
    - https://docs.nvidia.com/nsight-systems/UserGuide/index.html

Debugger.jl

- Install w/]
- In the REPL:
  - using Debugger
  - @enter fxn(X,y)
- Main commands:
  - n: step to next line
  - s: step into a function call
  - so: step out of the current function
  - bp add: add breakpoint
  - c: continue until breakpoint
  - bt: backtrace
  - fr: frame info (local variables by iteration)
  - q: quit

Infiltrator.jl

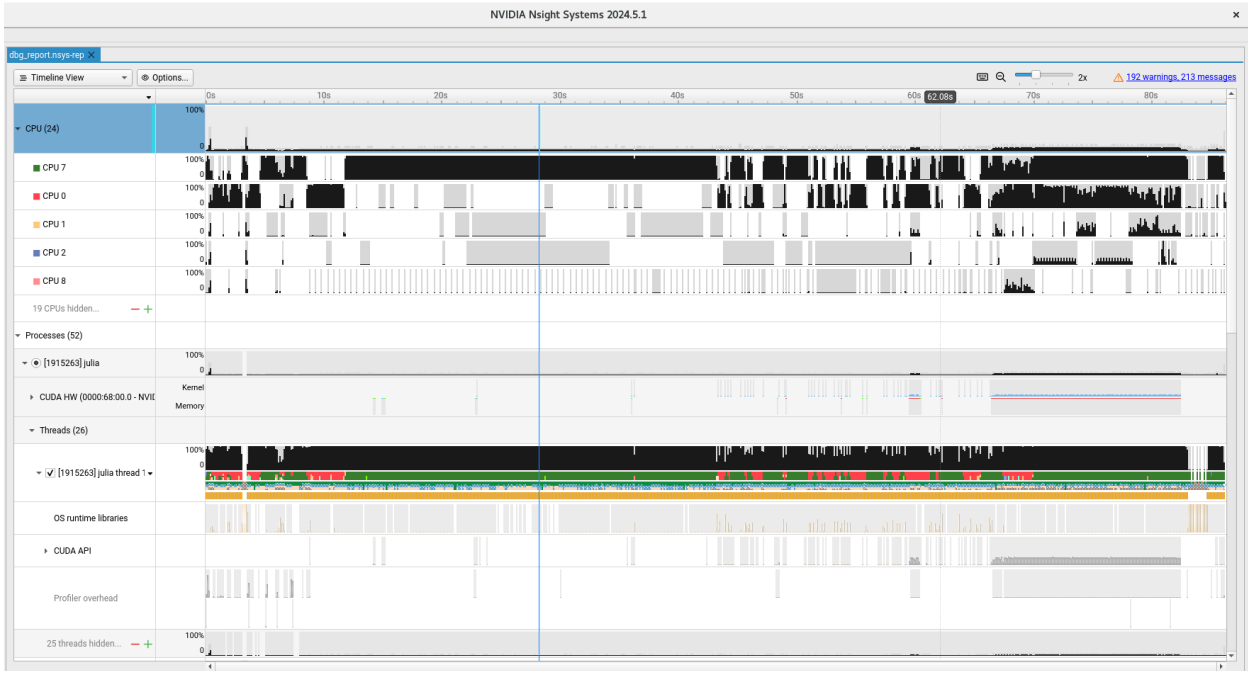
- Install w/]
- In your script:
  - using Infiltrator
  - @infiltrate can be added anywhere in your functions, ex. if ___, @infiltrate, end
- In the REPL:
  - @exfiltrate target_name actual_name
  - Then we can access this variable outside of the Infiltrator, via safehouse.target_name
- From there, we can manipulate and look at variables as needed

        - @trace
        - @locals
        - @exit
        - @continue

Nvidia Nsight Systems

- Installation
    - https://developer.nvidia.com/nsight-systems/get-started#reqs
    - ./NsightSystems-linux-public-2025.2.1.130-3569061.run
- Profiling
    - nsys profile -o result_profile julia script.jl
        - There are many many many more arguments you can add, tho!
        - https://docs.nvidia.com/nsight-systems/UserGuide/index.html
    - If running in a package environment:
        - julia --project=. -e 'using Pkg; Pkg.add("CUDA")'
        - nsys profile -o dbg_report julia --project=. script.jl
- Viewing Results
    - With GUI
        - Local:
            - nsys-ui result_profile.nsys-rep
        - Remote:
            - scp chans@kraken.iric.ca:~/result_profile.nsys-rep
            - ~/path-in-host-sys/nsight-sys result_profile.nsys-rep
    - Without GUI
        - nsys stats result_profile.nsys-rep
- Alternatively, we can run the profiler with the code (rather than post-run):
    - Local:
        - Launch application w/ ~/path/nsight-sys or clicking on it
        -
    - Remote:
        - Download the software onto your laptop:
            - https://developer.nvidia.com/nsight-systems/get-started#reqs
        - Launch application w/ w/ ~/path/nsight-sys or clicking on it
        - Connect via SSH

Example:



- CPU = CPU Utilization
    - Shows per-core activity across all CPU threads.
    - Black bars = active CPU execution.
    - Gray/white gaps = idle or context switching.
- Processes = GPU activity
    - Green bars = GPU kernels executing.
    - Orange/blue bars = memory transfers (ex. host to device)
- Also shows each Julia process + the threads
- NVTX section
    - CUDA.@nvtx "training loop" begin
      end
    - Allows u to have labelled code regions in the timeline
- Additional tabs for:
    - Analysis summary (CPU/GPU/software/etc. info)
    - Diagnostics summary (warnings, etc.)